

Attacking Your Two-Factor Authentication

(PS: Use Two-Factor Authentication)

08 Jun 2017 – K-LUG Technical Meeting – Rochester, MN

PRESENTED BY:

Vi Grey

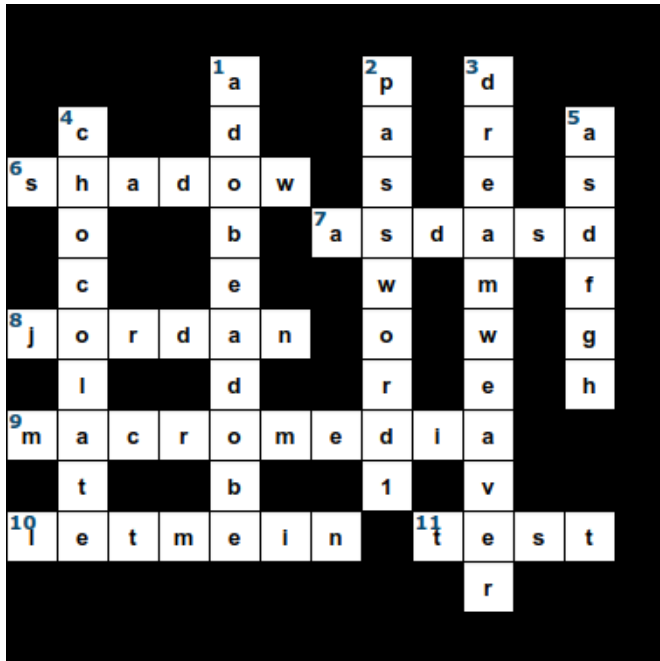
Independent Security Researcher

<https://vigrey.com>

Who Am I?

- Information Security Researcher
- Physical Security Enthusiast
- Software Engineer
- Focus of Study: Applied Cryptography, Exploit Development and Research
- Web Designer and Developer

A History Refresher



LILY HAY NEWMAN SECURITY 12.14.16 6:27 PM

HACK BRIEF: HACKERS BREACH A BILLION YAHOO ACCOUNTS. A BILLION

LinkedIn Lost 167 Million Account Credentials in Data Breach

Robert Hackett
May 18, 2016



Over 150 million breached records from Adobe hack have surfaced online

by Chris Welch | Nov 7, 2013, 6:08pm EST

A History Refresher - LinkedIn

- Account data breach happened in June 2012
- 167 million usernames and passwords stolen
- Users learned the extent of this data breach in 2016
- Passwords were improperly stored (unsalted SHA-1 hash, meaning any passwords that were the same provided the same stored data)
- Two-Factor Authentication was not available on LinkedIn at the time

A History Refresher - Yahoo

- Account data breach happened in August 2013
- Over 1 billion usernames and passwords stolen
- Users learned the extent of this data breach in 2016
- Passwords were stored in an insecure way (MD5 hash, although we don't know if the MD5 hashes were salted)

A History Refresher - Adobe

- Account data breach happened in September 2013
- Over 150 million usernames and passwords stolen
- Passwords were improperly stored (encrypted and using the same encryption key, meaning any passwords that were the same provided the same stored data)
- Password hints were stolen as well, making password decryption easier
- Two-Factor Authentication was not available on Adobe at the time

What is Two-Factor Authentication?

Two-Factor Authentication (2FA) is a second method of authentication, usually accompanying logging in with a username and password.

- First factor is something you know: Password
- Second factor is something you have: Phone, Security Hardware Token, Biometric Data, RFID Employee Badge, etc...

Why Use 2FA?

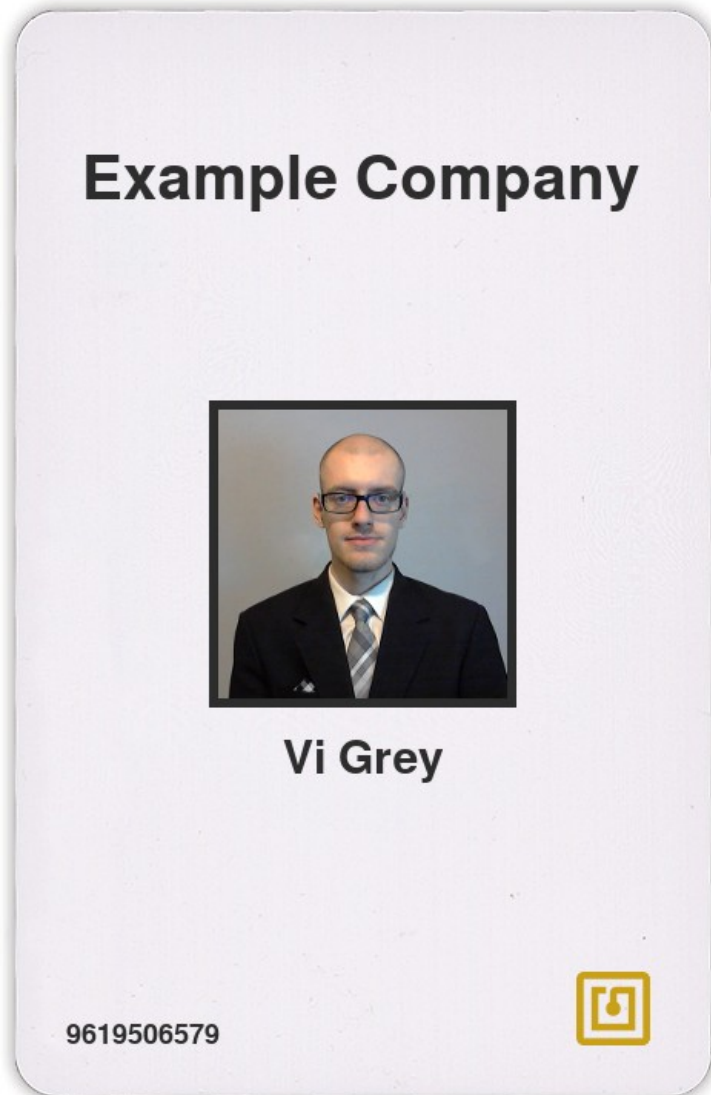
- Attackers can find or figure out usernames and passwords
 - Password Re-use is a BIG problem
 - Many services do not properly secure passwords before storing them
- Passwords can be entered by an attacker, often times remotely
- Attackers would need that second factor of authentication in order to log in as you

Forms of 2FA

- Static: Fingerprint, Retina, Many Employee RFID Badges
- Dynamic:
 - One Time Password: SMS*, RSA SecurID Hardware Token, Google Authenticator App
 - Challenge-Response: FIDO Universal Second Factor Hardware Token, Many Smart Cards

* The US National Institute of Standards and Technology (NIST) now recommends against SMS based 2FA in the draft of NIST Special Publication 800-63B

Static 2FA



What's Wrong with Static 2FA?

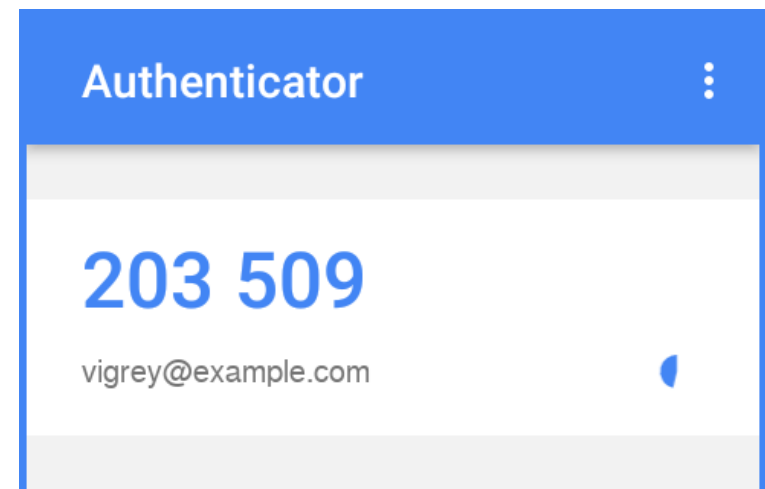
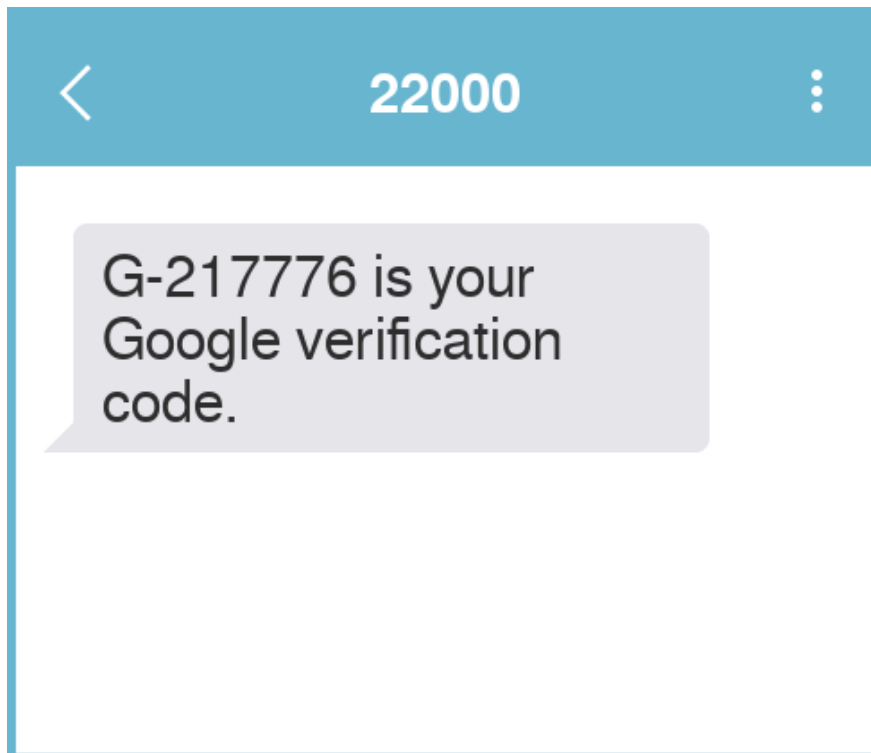
- Badge Cloning
- Static authentication tokens are predictable because they are supposed to be the same every time
- You can't change your biometric data, so if an attacker gains access to your biometric data, they have your authentication token forever
- Replay attacks

What Does Dynamic 2FA Do Better?

- Authentication tokens are unique every time
- Authentication tokens are unpredictable
- Replay attacks no longer work

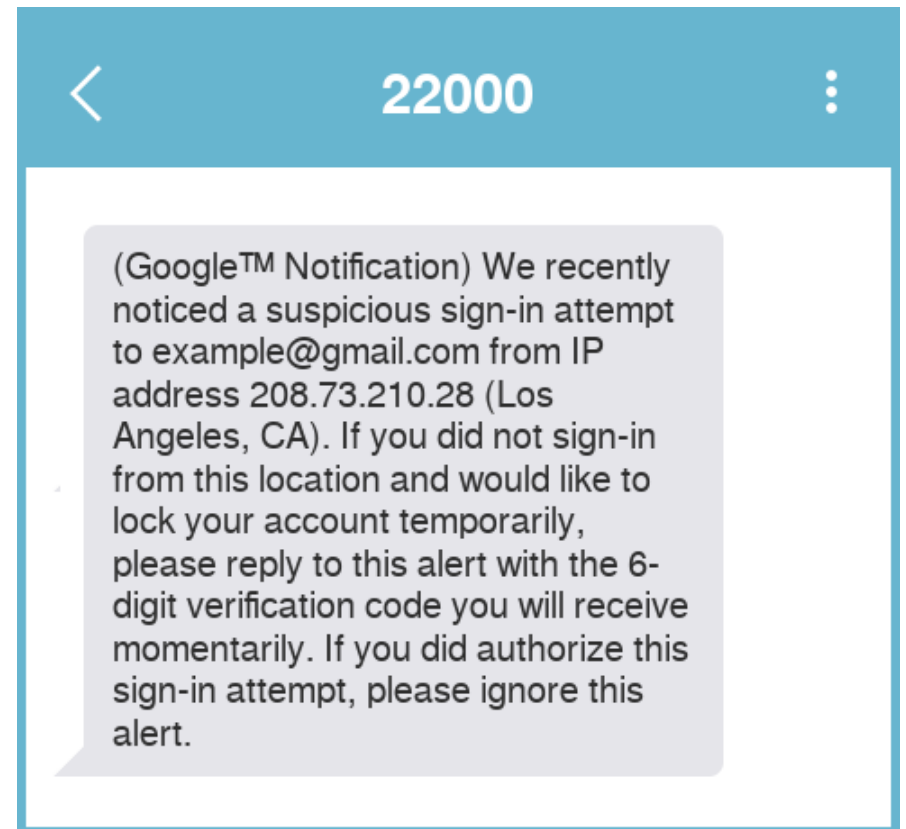
One Time Password (OTP)

- A one-time authentication token that is only valid for one use



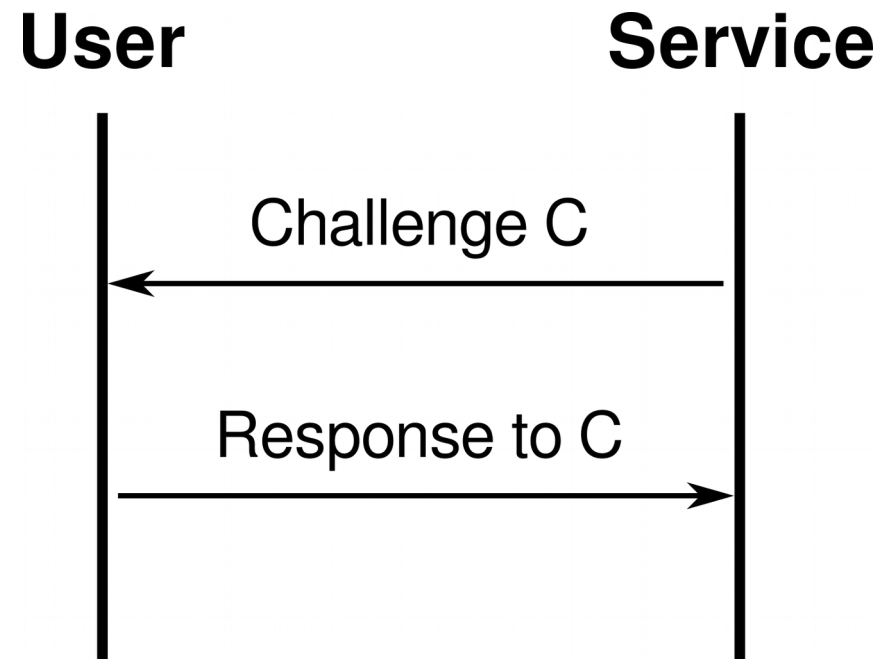
Attacking OTP

A properly worded phone call or text message can socially engineer a user, allowing an attacker to Man-in-the-Middle the authentication.



Challenge-Response

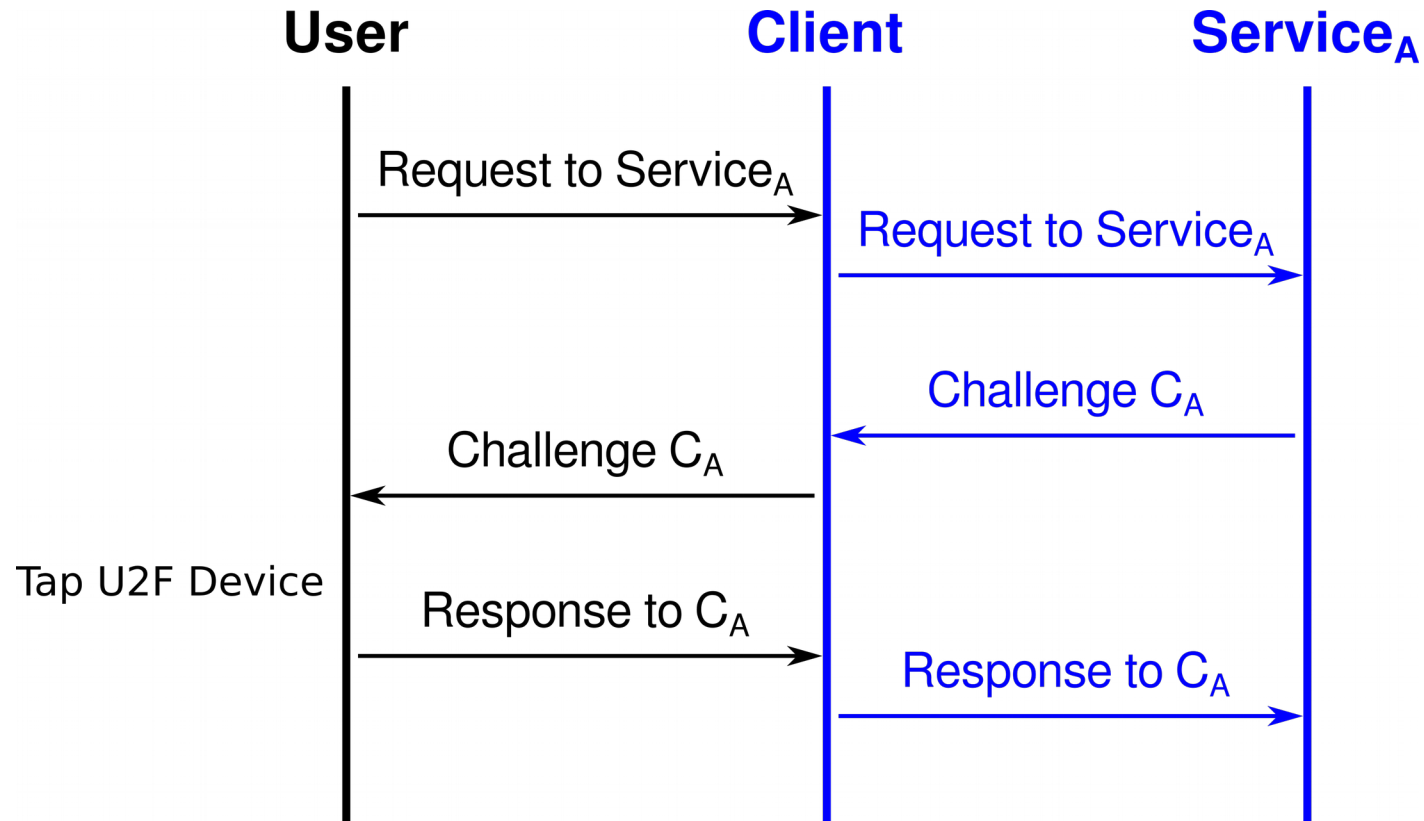
- A challenge is generated by the service and sent to the user to be processed into a Mathematical response that the service can verify



Universal Second Factor (U2F)

- The user sends a login request using a U2F client to a service
- The service sends a challenge to the client which sends the challenge to the U2F device
- The user interacts with their U2F device and a Mathematical response is sent to the client
- The client sends the response to the service to be verified

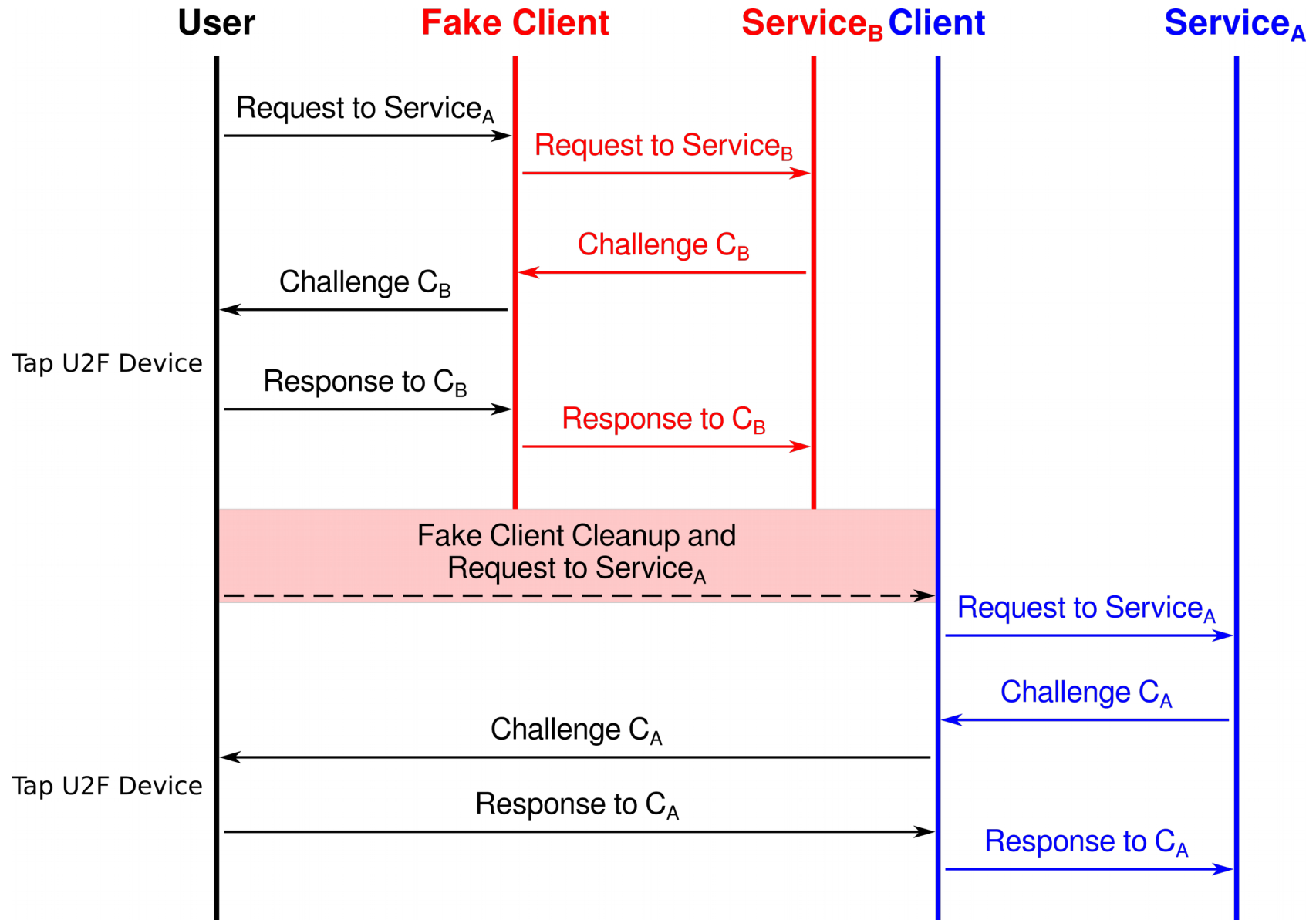
How U2F Works



Attacking U2F

- Most U2F devices do not have a display
- U2F devices trust the client
- The client communicates to U2F devices over the USB Human Interface Device (HID) specification, which is a computer industry standard
- The attacker can run a fake client before the real client runs
 - The fake client makes a request to Service_B that the user authenticates for the fake client before the real client makes a request to Service_A

Attacking U2F



U2F Attack Demo

Important Things to Note

- 2FA is another layer of security, but it's not perfect
- Using 2FA makes the attacker have to target the user, removing the ability to automate attacks, which is bad for economic viability

Questions?

vi@vigrey.com

vigrey.com

twitter.com/ViGreyInfoSec

github.com/ViGrey